

**IN THE CLAIMS:**

The following listing of claims will replace all prior listings of claims in the application:

1. (Previously Presented): A method of using a delegated connection table, comprising:
  - initializing an entry in the delegated connection table with connection state corresponding to a connection selected by a transmission control protocol (TCP) stack for processing by an offload unit;
  - updating the entry when a first frame is received for the connection;
  - parsing the first frame to extract TCP payload data;
  - uploading the TCP payload data to a memory; and
  - reading the entry when a second frame is transmitted for the connection.
2. (Previously Presented): The method of claim 1, further comprising updating the entry by copying a portion of the second frame into a portion of the entry in the delegated connection table when the second frame is transmitted.
3. (Cancelled)
4. (Previously Presented): The method of claim 1, further comprising uploading payload data to a location specified in the entry within a memory space of the memory that is allocated to an application program.
5. (Previously Presented): The method of claim 1, further comprising notifying the TCP stack when the TCP payload data of the first frame received is uploaded by the offload unit to at least one legacy buffer.
6. (Previously Presented): The method of claim 1, wherein the TCP payload data of the first frame is uploaded to a legacy buffer that is in a portion of the memory that is

allocated to a driver configured to interface between the offload unit and an application program.

7. (Previously Presented): The method of claim 1, further comprising:
  - receiving a third frame that does not correspond to another entry in the delegated connection table; and
  - uploading the third frame to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program.
8. (Original): The method of claim 6, wherein a sequence number in the first frame does not correspond to a sequence number stored in the delegated connection table.
9. (Cancelled)
10. (Previously Presented): The method of claim 1, further comprising uploading the payload data of the first frame to at least one legacy buffer that is in a first portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program when a user buffer in a second portion of the memory that is allocated to the application program is not available.
11. (Previously Presented): The method of claim 8, further comprising notifying [[an]]the application program to complete processing of the first frame.
12. (Previously Presented): The method of claim 8, further comprising uploading any subsequent frames received for the connection, to one or more additional legacy buffers, until resynchronization is signaled by the TCP stack.
13. (Original): The method of claim 12, wherein the resynchronization is accomplished by observing ACK numbers generated by the TCP stack.

14. (Original): The method of claim 12, further comprising:

determining an ACK number in a frame transmitted for the connection is more advanced than a sequence number stored in the entry; and  
copying the ACK number to the sequence number portion of the entry.

15. (Previously Presented): A method of accessing a delegated connection table during processing of a received frame, comprising:

reading a connection match portion of the delegated connection table;  
determining the received frame corresponds to an entry in the connection match portion of the delegated connection table;  
reading a connection data portion of the delegated connection table that stores an expected sequence number, an acknowledgment (ACK) number, timestamp data, and a count of unACKnowledged frames in the entry; and  
parsing the received frame to produce payload data.

16. (Original): The method of claim 15, further comprising:

modifying a portion of connection state data stored in the connection data portion of the delegated connection table.

17. (Previously Presented): The method of claim 15, further comprising:

reading a connection buffer portion of the delegated connection table to obtain user buffer information including a user buffer address and a corresponding user buffer length of a user buffer that is stored in a portion of memory allocated to an application program.

18. (Previously Presented): The method of claim 17, further comprising:

determining the user buffer information indicates a user buffer is not available; and  
requesting a user buffer by setting a request buffer flag in the connection buffer portion of the delegated connection table.

19. (Original): The method of claim 17, further comprising uploading the payload data to the user buffer.

20. (Previously Presented): The method of claim 18, further comprising:  
determining a receive buffer has reached a high water mark; and  
uploading the payload data to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the application program and an offload unit including the delegated connection table.

21. (Previously Presented): The method of claim 18, further comprising:  
determining a buffer request timer has expired; and  
uploading the payload data to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the application program and an offload unit including the delegated connection table.

22. (Previously Presented): A delegated connection table for storing delegated connection information, comprising:

a first storage resource configured to store user buffer information for delegated connections including a user buffer length and a user buffer address corresponding to a portion of memory that is allocated to an application program; and

a second storage resource configured to store delegated connection state information for the delegated connections including an expected sequence number, an acknowledgment (ACK) number, timestamp data, and a count of unACKnowledged frames.

23. (Previously Presented): The delegated connection table of claim 22, further comprising a third storage resource configured to store delegated connection identification information for the delegated connections including a destination IP

address, a source IP address, a source transmission control protocol (TCP) port, and a destination TCP port.

24. (Original): The delegated connection table of claim 22, further comprising a command processing unit configured to write to the first storage resource.
25. (Previously Presented): The delegated connection table of claim 22, further comprising a transmit engine configured to access the second storage resource and perform outbound frame processing.
26. (Previously Presented): The delegated connection table of claim 22, further comprising a receive engine configured to access the second storage resource and parse incoming frames and determine whether or not the incoming frames are valid.
27. (Original): The delegated connection table of claim 26, wherein the receive engine is configured to read the first storage resource.
28. (Original): The delegated connection table of claim 26, wherein the receive engine is configured to read the third storage resource.
29. (Previously Presented): The method of claim 1, wherein the updating of the entry when the first frame is received for the connection includes clearing an unACKnowledged count, updating an acknowledgment (ACK) number with a last ACKnowledged number, and updating a sequence number with an incremental sequence number that is stored in the entry.
30. (Previously Presented): The method of claim 16, wherein the modifying of the portion of the connection state data includes clearing an unACKnowledged count, updating the acknowledgment (ACK) number with a last ACKnowledged number, and updating the expected sequence number with an incremental sequence number.